

COPY PROTECTION WITHOUT NON-VOLATILE MEMORY

FIELD OF THE INVENTION

[0001] This invention relates generally to programmable logic devices and specifically to protecting circuit designs in programmable logic devices.

BACKGROUND OF THE INVENTION

[0002] Programmable logic devices (PLDs) are a well-known type of general purpose integrated circuit that may be programmed to implement one or more desired circuit designs. One type of PLD, a field programmable gate array (FPGA), typically includes an array of identical configurable logic blocks (CLBs) surrounded by input/output blocks (IOBs). Each CLB may be individually programmed to perform a variety of logic functions. A programmable interconnect selectively connects the CLBs to each other and to the various IOBs to implement complex logic functions and circuits.

[0003] To program the PLD to implement a circuit design, a user captures the circuit design using a well-known design capture tool, and then uses well-known software tools to convert the captured design into a device specific bitwise representation. The bitwise representation, commonly referred to as a configuration bitstream, is stored in a non-volatile storage device such as a Programmable Read Only Memory (PROM). Upon power-up, the non-volatile storage device transmits the configuration bitstream to the FPGA, where is loaded into a configuration memory array that controls various switches and multiplexers within the CLBs, IOBs, and programmable interconnect to implement the desired circuit design. Once configured, the FPGA implements the circuit design embodied in the configuration data. For a more detailed description of FPGA architecture, configuration, and operation, refer to The Xilinx 1998 Data

Book entitled "The Programmable Logic Data Book", Chapter 4, available from Xilinx, Inc., and incorporated herein by reference.

[0004] The FPGA's configuration memory array is typically a volatile memory such as Static Random Access Memory (SRAM). Thus, when the FPGA is powered off, the configuration data stored in its configuration memory array is lost. Consequently, when the FPGA is again powered up, it must be re-configured using configuration data provided by the non-volatile storage device, as described above.

[0005] The non-volatile storage device is typically external to the FPGA not only because of its size and cost but also because of the complexity of its manufacturing process. For example, the manufacturing process of a PROM requires several more masking and deposition steps than that of an SRAM, because while an SRAM uses cross-coupled transistors to store information, a PROM stores information using floating gate transistors, which require an additional layer of polysilicon. Because of the increased manufacturing complexity of non-volatile memory, as compared to volatile memory, the latest processing technology may be available for manufacturing volatile memory as much as a year or more before it is available for manufacturing non-volatile memory. Accordingly, providing the non-volatile memory external to the FPGA allows the FPGA to be manufactured using the latest manufacturing technology, which in turn provides a competitive advantage.

[0006] Because creation of the circuit design embodied in the configuration bitstream requires significant time and expense, it is desirable to protect the configuration data from illegal copying by competitors and unauthorized resellers. Unfortunately, the circuit design for an FPGA may be illegally copied by simply copying the configuration bitstream stored in the non-volatile memory. The copied bitstream may then be used to illegally configure other

FPGAs, or may be reverse-engineered to extract the circuit design.

[0007] To protect against unauthorized copying, the configuration data may be encrypted before it is stored in the non-volatile memory, transmitted to the FPGA in encrypted form, and then decrypted in the FPGA. Thus, in order to copy the circuit design embodied in the configuration data, one must determine the decryption key, thereby making unauthorized copying difficult. However, because the configuration data must be decrypted each time it is received from the external memory during power-up, the FPGA must include a non-volatile memory to store the decryption key. As discussed above, it is not desirable to include non-volatile memory within the FPGA. Accordingly, there is a need for a method of protecting the configuration data when transmitted to the FPGA that does not require a non-volatile memory within the FPGA.

SUMMARY OF THE INVENTION

[0008] A method and apparatus are disclosed that allow for the permanent storage of a decryption key within a PLD such as an FPGA without requiring non-volatile memory on the FPGA. In accordance with the present invention, an FPGA includes a plurality of configurable logic elements, a configuration circuit, a decryption circuit, and a fingerprint element. The fingerprint element generates a fingerprint that is indicative of inherent manufacturing process variations unique to the FPGA. The fingerprint is then used as a key for an encryption system that protects against illegal use and/or copying of the configuration data. In one embodiment, the fingerprint is used as an encryption key to encrypt the configuration data, which is then stored in an external non-volatile storage device such as a PROM. Upon power-up, the fingerprint element generates the fingerprint, and provides the fingerprint to the

decryption circuit. The storage device transmits the encrypted configuration data to the FPGA's decryption circuit, which in turn uses the fingerprint as a decryption key to decrypt the encrypted configuration data. The extracted configuration data may then be used to configure the configurable logic elements of the FPGA. Because the fingerprint key is generated each time the FPGA is powered up, the FPGA does not require non-volatile memory for its storage. Accordingly, embodiments of the present invention allow for encryption of the FPGA's configuration data without sacrificing the ability to fabricate the FPGA using the latest silicon manufacturing process technology.

[0009] The fingerprint element may generate the fingerprint in response to manufacturing process variations unique to the FPGA in any well-known manner. In some embodiments, differences in propagation delays of various circuit elements on the FPGA are used to generate the fingerprint. In one embodiment, frequency variations in oscillators resulting from differences in propagation delays are used to generate the fingerprint. In another embodiment, differences in transistor threshold voltages are used to generate the fingerprint. In yet another embodiment, variations in line widths are used to generate the fingerprint. In some embodiments, a ratio of measurable values may be used to generate the fingerprint to eliminate fingerprint drift resulting from temperature and voltage fluctuations.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] FIG. 1 is a block diagram of an FPGA including a fingerprint element in accordance with the present invention;

[0011] FIG. 2 is a flow chart illustrating configuration of an FPGA in accordance with one embodiment of the present invention;

[0012] FIG. 3 is a block diagram illustrating one embodiment of the fingerprint element of the present invention;

[0013] FIG. 4 is a block diagram illustrating another embodiment of the fingerprint element of the present invention;

[0014] FIG. 5 is a block diagram illustrating yet another embodiment of the fingerprint element of the present invention; and

[0015] FIG. 6 is a block diagram illustrating still another embodiment of the fingerprint element of the present invention.

[0016] Like components in the Figures are similarly labeled.

DETAILED DESCRIPTION OF THE INVENTION

[0017] FIG. 1 shows a circuit board 1 containing an FPGA 10 in accordance with the present invention that is connected to an encryption circuit 19 and an external storage device 20. The storage device 20 stores, in an encrypted form, configuration data that includes the bitwise representation of a desired circuit design that is to be implemented by the FPGA 10. The storage device 20 may be any non-volatile memory device, and in one embodiment is a PROM. The configuration data is provided to the encryption circuit 19, which in turn encrypts the configuration data in a well-known manner. The encryption circuit 19 provides the encrypted configuration data to the storage device 20 for storage therein. In some embodiments, the encryption circuit 19 uses a fingerprint generated by the fingerprint element 18 of the FPGA 10 as an encryption key to encrypt the configuration data, although other encryption keys may be used. Further, although shown in FIG. 1 as external to the storage device 20, in some embodiments, the encryption circuit 19 is included within the storage device 20.

[0018] The FPGA 10 includes a plurality of configurable logic elements 12, a configuration circuit 14, a decryption circuit 16, and a fingerprint element 18. The configurable logic elements 12 include well-known CLBs, IOBs, and a programmable interconnect (not shown for simplicity) that, when configured, implement the desired circuit design. In one embodiment, the configurable logic elements 12 are of the type used in the Virtex family of FPGAs available from Xilinx, Inc., 2100 Logic Drive, San Jose, California 95124, although other FPGA architectures may be used in accordance with the present invention. The configuration circuit 14 is conventional, and controls the loading of configuration data into the configurable logic elements 12. The decryption circuit 16 may be any well-known circuit for decrypting the encrypted configuration data received from the storage device 20.

[0019] During power-up of the board 1, the storage device 20 transmits the encrypted configuration data stored therein to the FPGA 10. The fingerprint element 18 generates a fingerprint that is indicative of inherent manufacturing process variations characteristic of the FPGA 10. In accordance with the present invention, the decryption circuit 16 uses the fingerprint as a decryption key to decrypt the encrypted configuration data to extract the (original) configuration data. Because of the unlikelihood that any two chips will have identical manufacturing process variations, the fingerprint generated by the fingerprint element 18 is unique to the FPGA 10. Thus, by using inherent manufacturing process variations unique to the FPGA 10 to generate the fingerprint (e.g., the decryption key) each time the FPGA 10 is powered up, the FPGA 10 does not need an on-board non-volatile memory to store the decryption key, thereby saving silicon area and cost. Further, because the fabrication of the FPGA 10 does not depend on whether a particular manufacturing process technology has been

developed for non-volatile memory, the FPGA 10 may be fabricated using the latest manufacturing process technology, thereby allowing for encryption of the FPGA 10's configuration data without sacrificing the competitive advantage realized by using the latest manufacturing process technology.

[0020] The competitive advantage realized by using the latest process technology, as opposed to using the previous generation process technology, is significant. For example, at present, while the latest available process for volatile memory is the 0.13 micron technology, the latest process available for non-volatile memory is the 0.25 micron technology. Since the 0.13 micron process technology produces chips that are about half as long and about half as wide as the 0.25 micron process technology, the chips produced by the 0.13 micron process technology are about one fourth the size, and therefore about one-fourth the cost, as chips produced by the 0.25 micron process technology. Thus, for example, by generating a decryption key within the FPGA 10 without using non-volatile memory, FPGAs incorporating embodiments of the present invention may be fabricated using 0.13 micron process technology. In contrast, FPGAs that include non-volatile memory to store the decryption key cannot presently be fabricated using 0.13 micron process technology, but rather must be fabricated using 0.25 micron process technology, which as discussed above is a competitive disadvantage.

[0021] Configuration of the FPGA 10 in one embodiment of the present invention is described below with reference to the flow chart of FIG. 2. During testing of the board 1, the FPGA 10 is powered up, and the fingerprint element 18 generates the fingerprint (step 21). The fingerprint is transmitted from the FPGA 10 to the encryption circuit 19 (step 22). Preferably, the fingerprint is transmitted from the FPGA 10 to the encryption circuit 19 in an encrypted

form using, for example, a well-known public-key cryptography system, although other cryptography systems may be used. The encryption circuit 19 receives configuration data that embodies the circuit design desired to be implemented by the FPGA 10, and uses the fingerprint as an encryption key to encrypt the configuration data (step 23). Any suitable encryption technique may be used. The encrypted configuration data is then provided to and stored in the external storage device 20 in a well-known manner (step 24).

[0022] To configure the FPGA 10 to implement the desired circuit design, the board 1 is powered up, and the FPGA 10 waits until the power supply becomes stable at a predetermined voltage. Then, a power-on reset step resets the configurable logic elements 12 (and other elements not shown for simplicity) in a well-known manner (step 25). The external storage device 20 transmits the encrypted configuration data to the decryption circuit 16 of the FPGA 10 (step 26). The fingerprint element 18 generates the fingerprint (step 27). As mentioned earlier, the fingerprint generated by the fingerprint element 18 is a multi-bit binary representation of one or more inherent manufacturing process variations unique to the FPGA 10, and may be of any length. In one embodiment, the fingerprint is 150 bits.

[0023] The fingerprint element 18 provides the fingerprint to the decryption circuit 16, which in turn uses the fingerprint as a decryption key to decrypt the encrypted configuration data to extract the (original) configuration data (step 28). The configuration data is then provided to the configuration circuit 14, which uses the configuration data to program the configurable logic elements 12 to implement the desired circuit design (step 29). In this manner, in order to appropriate the configuration data, a person must determine the decryption key, thereby making

illegal copying and/or use of the configuration data difficult.

[0024] The fingerprint element 18 may be any well-known circuit or software tool that generates a fingerprint indicative of manufacturing process variations unique to the FPGA 10. In some embodiments, differences in transistor threshold voltages caused by such process variations are used to generate the fingerprint. In one embodiment, the fingerprint element 18 includes a plurality of pairs of transistors, where differences in the threshold voltage between the transistors within each pair are used to generate a corresponding bit of the fingerprint.

[0025] Referring to FIG. 3, the fingerprint element 30, which is one embodiment of the fingerprint element 18, includes a plurality of transistor pairs 32a(1)-32b(1) to 32a(n)-32b(n) connected between a power supply V_{DD} and a sensing circuit 34. A read voltage V_R having a magnitude within the range of transistor threshold voltage variations is applied to the gates of each of the transistor pairs 32a(1)-32b(1) to 32a(n)-32b(n). Because the transistors 32a and 32b of each pair have different threshold voltages, when V_R is applied thereto, one of the transistors in each pair turns on earlier than the other transistor in the pair. The sensing circuit 34 determines which transistor 32a or 32b of each transistor pair turns on earlier by, for example, detecting current flow in the transistors 32a and 32b, and in response thereto generates a corresponding bit of the n-bit fingerprint. For example, in one embodiment, for each of transistor pairs 32a(1)-32b(1) to 32a(n)-32b(n), the sensing circuit 34 generates a fingerprint bit of logic 1 if the first transistor 32a of the pair turns on before the second transistor 32b of the pair and, conversely, generates a fingerprint bit of logic 0 if the second transistor 32b of the pair turns on before the first transistor 32a of the pair. The fingerprint bits generated from corresponding

transistor pairs 32a(1)-32b(1) to 32a(n)-32b(n) are combined (e.g., concatenated) to generate the n-bit fingerprint.

[0026] In some embodiments, the read voltage V_R is slowly ramped from a first voltage to a second voltage, where the first voltage is less than all the threshold voltages of the transistor pairs 32a(1)-32b(1) to 32a(n)-32b(n), and the second voltage is greater than all the threshold voltages of the transistor pairs 32a(1)-32b(1) to 32a(n)-32b(n).

[0027] Because the manufacturing process variations that result in variations in transistor threshold voltages are random, the fingerprint generated by the fingerprint element 30 is unique to the FPGA 10. However, because transistor threshold voltages gradually vary over time, the fingerprint generated by the fingerprint element 30 may undesirably drift (i.e., change) over time. If allowed to drift, the fingerprint may no longer be able to decrypt the configuration data received from the external storage device 20.

[0028] Fingerprint drift may be avoided by generating the fingerprint according to manufacturing process variations that do not change over time. In one embodiment, the fingerprint element 18 includes a plurality of line segments whose deviation from a predetermined line width is used to generate corresponding bits of the fingerprint. For example, FIG. 4 shows a fingerprint element 40 including a plurality of line segments 42(1)-42(n) connected to a sensing circuit 44. For each line segment 42(1)-42(n), the sensing circuit 44 determines whether its line width is less than the predetermined line width, and generates a corresponding bit of the fingerprint accordingly. For example, if the line width of a line segment 42 is less than the predetermined line width, then the sensing circuit 44 generates a corresponding fingerprint bit of logic 0, and conversely, if the line width of the line segment 42 is not less than the predetermined line width, then the sensing circuit 44 generates a corresponding fingerprint bit of

logic 1. The sensing circuit 44 may determine the relative line width of line segments 42(1)-42(n) using any well-known electrical technique. The statistical variation of the line widths may be maximized by fabricating the line widths as narrow as possible. Because line widths do not vary over time, the fingerprint generated by the fingerprint element 40 remains constant over time.

[0029] In other embodiments, propagation delays of circuit elements on the FPGA 10 are measured and used to generate the fingerprint in a well-known manner. In one embodiment, an oscillator within the FPGA 10 is used to generate the fingerprint. FIG. 5 shows a fingerprint element 50 including an oscillator 52 and a sensing circuit 54. Because transistor delays are sensitive to manufacturing process variations, which as described above are random, the specific frequency of the fingerprint oscillator 52 on the FPGA 10 is unique. To generate the fingerprint, power is applied to the oscillator 52, and the number of times that the oscillator 52 switches during a predetermined time interval is counted. The resultant binary equivalent of the oscillator count is then used as the fingerprint. Reference for the predetermined time interval, which may be of any suitable duration, may be provided by well-known delay locked loops (DLL) within the configurable logic elements 12 of the FPGA 10 (see also FIG. 1). Thus, although shown in FIG. 5 as internal to the fingerprint element 50, in other embodiments, the oscillator 52 may be external to the fingerprint element 50, e.g., the oscillator 52 is fabricated in or implemented by the configurable logic elements 12 (see also FIG. 1). In some embodiments, the oscillator 52 is implemented within a CLB by configuring the CLB as an inverter with feedback. In such embodiments, the CLB may be within either the fingerprint element 50 or the configurable logic elements 12.

[0030] However, because transistor delays are sensitive to temperature and voltage fluctuations, the frequency of the oscillator 52 may be sensitive to temperature and voltage fluctuations. As a result, the fingerprint element 50 may not generate the same fingerprint when the FPGA 10 is exposed to significant temperature or voltage fluctuations, thereby resulting in fingerprint drift. Accordingly, the following embodiments provide immunity from fingerprint drift resulting from temperature and voltage fluctuations.

[0031] In one embodiment, a ratio of oscillator counts is used to generate the fingerprint in order to eliminate the effect of temperature and voltage fluctuations. For example, FIG. 6 shows a fingerprint element 60 including first and second oscillators 62 and 64 connected to a sensing circuit 66. When power is applied, the sensing circuit 66 counts the number of times that each oscillator 62 and 64 switches during the predetermined time interval. Reference for the predetermined time interval may be provided by a DLL as described above with respect to the fingerprint element 50. The sensing circuit 66 divides the count value of one oscillator by the count value of the other oscillator to generate a ratio, the binary equivalent of which is used as the fingerprint. Because the respective frequencies of oscillators 62 and 64 vary equally over temperature and voltage fluctuations, the ratio of their resultant count values, and therefore the fingerprint generated therefrom, remains constant over temperature and voltage fluctuations. Although shown in FIG. 6 as internal to the fingerprint element 60, in other embodiments the oscillators 62 and 64 may be external to the fingerprint element 60. In some embodiments, the oscillators 62 and 64 are implemented using CLBs, as described above.

[0032] Referring again to FIG. 5, in another embodiment, the sensing circuit 50 is configured to ignore the m least significant bits of the n-bit oscillator count value

provided from the oscillator 52, with the remaining n-m bits of the oscillator count value being used as the fingerprint. By truncating the m least significant bits of the oscillator count value, the resultant shortened fingerprint represents a range of oscillator count values, and therefore includes a built-in tolerance for frequency variations resulting from temperature and voltage fluctuations. However, because sensing circuit 54 must be pre-configured to ignore the m least significant bits when generating the fingerprint, the frequency range represented by shortened fingerprint is fixed. Further, if the frequency range represented by the shortened fingerprints of two or more different FPGAs 10 overlap, then the resulting decryption keys may be same, which in turn diminishes security.

[0033] While particular embodiments of the present invention have been shown and described, it will be obvious to those skilled in the art that changes and modifications may be made without departing from this invention in its broader aspects and, therefore, the appended claims encompass within their scope all such changes and modifications as fall within the true spirit and scope of this invention. For example, although in the above examples the configuration data is described as being stored in encrypted format in the storage device 20, in other embodiments, the configuration data may be stored in the storage device 20 in un-encrypted form. In such embodiments, the storage device 20 also includes a well-known encryption circuit that encrypts the configuration data using the fingerprint as the encryption key prior to its transmission to the FPGA 10. Further, although described above as using the fingerprint for both the decryption key and the encryption key, in actual embodiments, a key other than the fingerprint may be used as the encryption key.